

Finitary Languages

Krishnendu Chatterjee¹ and Nathanaël Fijalkow^{1,2}

¹ IST Austria (Institute of Science and Technology, Austria)
krishnendu.chatterjee@ist.ac.at

² ÉNS Cachan (École Normale Supérieure de Cachan, France)
nathanael.fijalkow@gmail.com

Abstract. The class of ω -regular languages provides a robust specification language in verification. Every ω -regular condition can be decomposed into a safety part and a liveness part. The liveness part ensures that something good happens “eventually”. Finitary liveness was proposed by Alur and Henzinger as a stronger formulation of liveness [AH98]. It requires that there exists an unknown, fixed bound b such that something good happens within b transitions. In this work we consider automata with finitary acceptance conditions defined by finitary Büchi, parity and Streett languages. We give their topological complexity of acceptance conditions, and present a regular-expression characterization of the languages they express. We provide a classification of finitary and classical automata with respect to the expressive power, and give optimal algorithms for classical decisions questions on finitary automata. We (a) show that the finitary languages are Σ_2 -complete; (b) present a complete picture of the expressive power of various classes of automata with finitary and infinitary acceptance conditions; (c) show that the languages defined by finitary parity automata exactly characterize the star-free fragment of ωB -regular languages; and (d) show that emptiness is NLOGSPACE-complete and universality as well as language inclusion are PSPACE-complete for finitary automata.

1 Introduction

Classical ω -regular languages: strengths and weakness. The class of ω -regular languages provides a robust language for specification for solving control and verification problems (see, *e.g.*, [PR89,RW87]). Every ω -regular specification can be decomposed into a safety part and a liveness part [AS85]. The safety part ensures that the component will not do anything “bad” (such as violate an invariant) within any finite number of transitions. The liveness part ensures that the component will do something “good” (such as proceed, or respond, or terminate) in the long-run. Liveness can be violated only in the limit, by infinite sequences of transitions, as no bound is stipulated on when the “good” thing must happen. This infinitary, classical formulation of liveness has both strengths and weaknesses. A main strength is robustness, in particular, independence from the chosen granularity of transitions. Another main strength is simplicity, allowing

liveness to serve as an abstraction for complicated safety conditions. For example, a component may always respond in a number of transitions that depends, in some complicated manner, on the exact size of the stimulus. Yet for correctness, we may be interested only that the component will respond “eventually”. However, these strengths also point to a weakness of the classical definition of liveness: it can be satisfied by components that in practice are quite unsatisfactory because no bound can be put on their response time.

Stronger notion of liveness. For the weakness of the infinitary formulation of liveness, alternative and stronger formulations of liveness have been proposed. One of these is *finitary* liveness [AH98]: finitary liveness does not insist on a response within a known bound b (*i.e.*, every stimulus is followed by a response within b transitions), but on response within some unknown bound (*i.e.*, there exists b such that every stimulus is followed by a response within b transitions). Note that in the finitary case, the bound b may be arbitrarily large, but the response time must not grow forever from one stimulus to the next. In this way, finitary liveness still maintains the robustness (independence of step granularity) and simplicity (abstraction of complicated safety) of traditional liveness, while removing unsatisfactory implementations.

Finitary parity and Streett conditions. The classical infinitary notion of fairness is given by the Streett condition: it consists of a set of d pairs of requests and corresponding responses (grants) and requires that every request that appears infinitely often must be responded infinitely often. Its finitary counterpart, the finitary Streett condition requires that there is a bound b such that in the limit every request is responded within b steps. The classical infinitary parity condition consists of a priority function and requires that the minimum priority visited infinitely often is even. Its finitary counterpart, the finitary parity condition requires that there is a bound b such that in the limit after every odd priority a lower even priority is visited within b steps.

Results on classical automata. There are several robust results on the languages expressible by automata with infinitary Büchi, parity and Streett conditions, as follows: (a) *Topological complexity*: it is known that Büchi languages are Π_2 -complete, whereas parity and Streett languages lie in the boolean closure of Σ_2 and Π_2 [MP92]; (b) *Automata expressive power*: non-deterministic automata with Büchi conditions have the same expressive power as deterministic and non-deterministic parity and Streett automata [Cho74,Saf92]; and (c) *Regular-expression characterization*: the class of languages expressed by deterministic parity is exactly defined by ω -regular expressions (see the handbook [Tho97] for details).

Our results. For finitary Büchi, parity and Streett languages, topological, automata-theoretic, regular-expression and decision problems studies were all missing. In this work we present results in the four directions, as follows:

1. *Topological complexity.* We show that finitary Büchi, parity and Streett conditions are Σ_2 -complete.
2. *Automata expressive power.* We show that finitary automata are incomparable in expressive power with classical automata. As in the infinitary setting, we show that non-deterministic automata with finitary Büchi, parity and Streett conditions have the same expressive power, as well as deterministic parity and Streett automata, which are strictly more expressive than deterministic finitary Büchi automata. However, in contrast to the infinitary case, for finitary parity condition, non-deterministic automata are strictly more expressive than the deterministic counterpart. As a by-product we derive boolean closure properties for finitary automata.
3. *Regular-expression characterization.* We consider the characterization of finitary automata through an extension of ω -regular languages defined as ωB -regular languages by [BC06]. We show that languages defined by non-deterministic finitary Büchi automata are exactly the star-free fragment of ωB -regular languages.
4. *Decision problems.* We show that emptiness is NLOGSPACE-complete and universality as well as language inclusion are PSPACE-complete for finitary automata.

Related works. The notion of finitary liveness was proposed and studied in [AH98], and games with finitary objectives was studied in [CHH09]. A generalization of ω -regular languages as ωB -regular languages was introduced in [BC06] and variants have been studied in [BT09] (also see [Boj10] for a survey); a topological characterization has been given in [HST10]. Our work along with topological and automata-theoretic studies of finitary languages, explores the relation between finitary languages and ωB -regular expressions, rather than identifying a subclass of ωB -regular expressions. We identify the exact subclass of ωB -regular expressions that corresponds to non-deterministic finitary parity automata.

2 Definitions

2.1 Languages topological complexity

Let Σ be a finite set, called the alphabet. A word w is a sequence of letters, which can be either finite or infinite. A language is a set of words: $L \subseteq \Sigma^*$ is a language over finite words and $L \subseteq \Sigma^\omega$ over infinite words.

Cantor topology and Borel hierarchy. Cantor topology on Σ^ω is given by *open* sets: a language is open if it can be described as $W \cdot \Sigma^\omega$ where $W \subseteq \Sigma^*$. Let Σ_1 denote the open sets and Π_1 denote the closed sets (a language is closed if its complement is open): they form the first level of the Borel hierarchy. Inductively, we define: Σ_{i+1} is obtained as countable union of Π_i sets; and Π_{i+1} is obtained as countable intersection of Σ_i sets. The higher a language is in the Borel hierarchy, the higher its topological complexity.

Since the above classes are closed under continuous preimage, we can define the notion of Wadge reduction [Wad84]: L reduces to L' , denoted by $L \preceq L'$, if there exists a continuous function $f : \Sigma^\omega \rightarrow \Sigma^\omega$ such $L = f^{-1}(L')$, where $f^{-1}(L')$ is the preimage of L' by f . A language is hard with respect to a class if all languages of this class reduce to it. If it additionally belongs to this class, then it is complete.

For $L \subseteq \Sigma^\omega$, let $\text{pref}(L) \subseteq \Sigma^*$ be the set of finite prefixes of words in L . The following property holds:

Proposition 1. *For all languages $L \subseteq \Sigma^\omega$, L is closed if and only if, for all infinite words w , if all finite prefixes of w are in $\text{pref}(L)$, then $w \in L$.*

Classical liveness conditions. We now consider three classes of languages that are widespread in verification and specification. They define liveness properties, *i.e.*, intuitively say that something good will happen “eventually”. For an infinite word w , let $\text{Inf}(w) \subseteq \Sigma$ denote the set of letters that appear infinitely often in w . The class of Büchi languages is defined as follows, given $F \subseteq \Sigma$:

$$\text{Büchi}(F) = \{w \mid \text{Inf}(w) \cap F \neq \emptyset\}$$

i.e., the Büchi condition requires that some letter in F appears infinitely often. The class of parity languages is defined as follows, given $p : \Sigma \rightarrow \mathbb{N}$ a priority function that maps letters to integers (representing priorities):

$$\text{Parity}(p) = \{w \mid \min(p(\text{Inf}(w))) \text{ is even}\}$$

i.e., the parity condition requires that the lowest priority the appears infinitely often is even. The class of Streett languages is defined as follows, given $(R, G) = (R_i, G_i)_{1 \leq i \leq d}$, where $R_i, G_i \subseteq \Sigma$ are request-grant pairs:

$$\text{Streett}(R, G) = \{w \mid \forall i, 1 \leq i \leq d, \text{Inf}(w) \cap R_i \neq \emptyset \Rightarrow \text{Inf}(w) \cap G_i \neq \emptyset\}$$

i.e., the Streett condition requires that for all requests R_i , if it appears infinitely often, then the corresponding grant G_i also appears infinitely often.

The following theorem presents the topological complexity of the classical languages:

Theorem 1 (Topological complexity of classical languages [MP92]).

- For all $\emptyset \subset F \subset \Sigma$, the language $\text{Büchi}(F)$ is Π_2 -complete.
- The parity and Streett languages lie in the boolean closure of Σ_2 and Π_2 .

2.2 Finitary languages

The finitary parity and Streett languages have been defined in [CHH09]. We recall their definitions, and also specialize them to finitary Büchi languages. Let $(R, G) = (R_i, G_i)_{1 \leq i \leq d}$, where $R_i, G_i \subseteq \Sigma$, the definition for $\text{FinStreett}(R, G)$ uses distance sequence as follows:

$$\text{dist}_k^j(w, (R, G)) = \begin{cases} 0 & w_k \notin R_j \\ \inf\{k' - k \mid k' \geq k, w_{k'} \in G_j\} & w_k \in R_j \end{cases}$$

i.e., given a position k where R_j is requested, $\text{dist}_k^j(w, (R, G))$ is the time steps (number of transitions) between the request R_j and the corresponding grant G_j . Note that $\inf(\emptyset) = \infty$. Then $\text{dist}_k(w, (R, G)) = \max\{\text{dist}_k^j(w, p) \mid 1 \leq j \leq d\}$ and:

$$\text{FinStreett}(R, G) = \{w \mid \limsup_k \text{dist}_k(w, (R, G)) < \infty\}$$

i.e., the finitary Streett condition requires the supremum limit of the distance sequence to be bounded.

Since parity languages can be considered as a particular case of Streett languages, where $G_1 \subseteq R_1 \subseteq G_2 \subseteq R_2 \dots$, the latter allows to define $\text{FinParity}(p)$. The same applies to finitary Büchi languages, which is a particular case of finitary parity languages where the letters from the set F have priority 0 and others have priority 1. We get the following definitions. Let $p : \Sigma \rightarrow \mathbb{N}$ a priority function, we define:

$$\text{dist}_k(w, p) = \inf\{k' - k \mid k' \geq k, p(w_{k'}) \text{ is even and } p(w_{k'}) \leq p(w_k)\}$$

i.e., given a position k where $p(w_k)$ is odd, $\text{dist}_k(w, p)$ is the time steps between the odd priority $p(w_k)$ and a lower even priority. Then $\text{FinParity}(p) = \{w \mid \limsup_k \text{dist}_k(w, p) < \infty\}$. We define similarly the finitary Büchi language: given $F \subseteq \Sigma$, let:

$$\text{next}_k(w, F) = \inf\{k' - k \mid k' \geq k, w_{k'} \in F\}$$

i.e., $\text{next}_k(w, F)$ is the time steps before visiting a letter in F . Then $\text{FinBüchi}(F) = \{w \mid \limsup_k \text{next}_k(w, F) < \infty\}$.

2.3 Automata, ω -regular and finitary languages

Definition 1. An automaton is a tuple $\mathcal{A} = (Q, \Sigma, Q_0, \delta, \text{Acc})$, where Q is a finite set of states, Σ is the finite input alphabet, $Q_0 \subseteq Q$ is the set of initial states, $\delta \subseteq Q \times \Sigma \times Q$ is the transition relation and $\text{Acc} \subseteq Q^\omega$ is the acceptance condition.

An automaton is deterministic if it has a single initial state and for every state and letter there is at most one transition. The transition relation of deterministic automata are described by functions $\delta : Q \times \Sigma \rightarrow Q$. An automaton is complete if for every state and letter there is a transition. This is the case when the transition function is *total*.

Runs. A run $\rho = q_0 q_1 \dots$ is a word over Q , where $q_0 \in Q_0$. The run ρ is accepting if it is infinite and $\rho \in \text{Acc}$. We will write $p \xrightarrow{a} q$ to denote $(p, a, q) \in \delta$. An infinite word $w = w_0 w_1 \dots$ induces possibly several runs of \mathcal{A} : a word w induces a run $\rho = q_0 q_1 \dots$ if for all $n \in \mathbb{N}$, $q_n \xrightarrow{w_n} q_{n+1} \dots$. The language accepted by \mathcal{A} , denoted by $\mathcal{L}(\mathcal{A}) \subseteq \Sigma^\omega$, is:

$$\mathcal{L}(\mathcal{A}) = \{w \mid \text{there exists an accepting run } \rho \text{ induced by } w\}.$$

Acceptance conditions. We will consider various acceptance conditions for automata obtained from the last section by considering Q as the alphabet. For example, given $F \subseteq Q$, the languages $\text{Büchi}(F)$ and $\text{FinBüchi}(F)$ define Büchi and finitary Büchi acceptance conditions, respectively. Automata with finitary acceptance conditions are referred as finitary automata; classical automata are those equipped with infinitary acceptance conditions.

Notation 1 We use a standard notation to denote the set of languages recognized by some class of automata. The first letter is either N or D , where N stands for “non-deterministic” and D stands for “deterministic”. The last letter refers to the acceptance condition: B stands for “Büchi”, P stands for “parity” and S stands for “Streett”. The acceptance condition may be prefixed by F for “finitary”. For example, NP denotes non-deterministic parity automata, and DFS denotes deterministic finitary Streett automata. We have the following combination:

$$\left\{ \begin{matrix} N \\ D \end{matrix} \right\} \cdot \left\{ \begin{matrix} F \\ \varepsilon \end{matrix} \right\} \cdot \left\{ \begin{matrix} B \\ P \\ S \end{matrix} \right\}$$

We denote by \mathbb{L}_ω the class of languages accepted by deterministic parity automata. The following theorem summarizes the results of expressive power of classical automata [Büc62,Saf92,Cho74,GH82]:

Theorem 2 (Expressive power results for classical automata).

$$DB \subset \mathbb{L}_\omega \doteq NB = DP = NP = DS = NS$$

3 Topological complexity

In this section we define a finitary operator UniCloOmg that allows us to describe finitary Büchi, finitary parity and finitary Streett languages topologically and to relate them to the classical Büchi, parity and Streett languages; we then give their topological complexity.

Union-closed-omega-regular operator on languages. Given a language $L \subseteq \Sigma^\omega$, the language $\text{UniCloOmg}(L) \subseteq \Sigma^\omega$ is the *union* of the languages M that are subsets of L , ω -regular and *closed*, i.e., $\text{UniCloOmg}(L) = \bigcup \{M \mid M \subseteq L, M \in \Pi_1, M \in \mathbb{L}_\omega\}$.

Proposition 2. *For all languages $L \subseteq \Sigma^\omega$ we have $\text{UniCloOmg}(L) \in \Sigma_2$.*

Proof. Since the set of finite automata can be enumerated in sequence, it follows that \mathbb{L}_ω is countable. So for all languages L , the set $\text{UniCloOmg}(L)$ is described as a countable union of closed sets. Hence $\text{UniCloOmg}(L) \in \Sigma_2$. ■

We present a *pumping lemma* for ω -regular languages that we will use to prove the topological complexity of finitary languages.

Lemma 1 (A pumping lemma). *Let M be an ω -regular language. There exists n_0 such that for all words $w \in M$, for all positions $j \geq n_0$, there exist $j \leq i_1 < i_2 \leq j + n_0$ such that for all $\ell \geq 0$ we have $w_0 w_1 w_2 \dots w_{i_1-1} \cdot (w_{i_1} w_{i_1+1} \dots w_{i_2-1})^\ell \cdot w_{i_2} w_{i_2+1} \dots \in M$.*

Proof. Given M is a ω -regular language, let \mathcal{A} be a complete and deterministic parity automata that recognizes M , and let n_0 be the number of states of \mathcal{A} . Consider a word $w = w_0 w_1 w_2 \dots$ such that $w \in M$, and let $\rho = q_0 q_1 q_2 \dots$ be the unique run induced by w in \mathcal{A} . Consider a position j in w such that $j \geq n_0$. Then there exist $j \leq i_1 < i_2 \leq j + n_0$ such that $q_{i_1} = q_{i_2}$, this must happen as \mathcal{A} has n_0 states. For $\ell \geq 0$, if we consider the word $w^\ell = w_0 w_1 w_2 \dots w_{i_1-1} \cdot (w_{i_1} w_{i_1+1} \dots w_{i_2-1})^\ell \cdot w_{i_2} w_{i_2+1} \dots$, then the unique run induced by w^ℓ in \mathcal{A} is $\rho^\ell = q_0 q_1 q_2 \dots q_{i_1-1} \cdot (q_{i_1} q_{i_1+1} \dots q_{i_2-1})^\ell \cdot q_{i_2} q_{i_2+1} \dots$. Since the parity condition is independent of finite prefixes and the run ρ is accepted by \mathcal{A} , it follows that ρ^ℓ is accepted by \mathcal{A} . Since \mathcal{A} recognizes M , we have $w^\ell \in M$. ■

The following lemma shows that $\text{FinStreett}(R, G)$ is obtained by applying the UniCloOmg operator to $\text{Streett}(R, G)$.

Lemma 2. For all $(R, G) = (R_i, G_i)_{1 \leq i \leq d}$, where $R_i, G_i \subseteq \Sigma$, we have

$$\text{UniCloOmg}(\text{Streett}(R, G)) = \text{FinStreett}(R, G).$$

Proof. We present the two directions of the proof.

1. We first show that $\text{UniCloOmg}(\text{Streett}(R, G)) \subseteq \text{FinStreett}(R, G)$. Let $M \subseteq \text{Streett}(R, G)$ such that M is closed and ω -regular. Let $w = w_0 w_1 \dots \in M$, and assume towards contradiction, that $\limsup_k \text{dist}_k(w, (R, G)) = \infty$. Hence for all $n_0 \in \mathbb{N}$, there exists $n \in \mathbb{N}$ such that $n \geq n_0$ and $\text{dist}_n(w, (R, G)) \geq n_0$. Let $n_0 \in \mathbb{N}$ given by the pumping lemma on M , from above given n_0 we obtain j such that $j \geq n_0$ and $\text{dist}_j(w, (R, G)) \geq n_0$. By the pumping lemma we obtain the witness $j \leq i_1 < i_2 \leq j + n_0$. Let $u = w_0 w_1 \dots w_{i_1-1}$, $v = w_{i_1} w_{i_1+1} \dots w_{i_2-1}$ and $w' = w_{i_2} w_{i_2+1} \dots$. Since $w \in M$, by the pumping lemma for all $\ell \geq 0$ we have $uv^\ell w' \in M$. This entails that all finite prefixes of the infinite word uv^ω are in $\text{pref}(M)$. Since M is closed, it follows that $uv^\omega \in M$. Since $\text{dist}_j(w, (R, G)) \geq n_0$ it follows that there is some request i in position j , and there is no corresponding grant i for the next n_0 steps. Hence there is a position j' in v such that there is request i at j' and no corresponding grant in v , and thus it follows that the word $uv^\omega \notin \text{Streett}(R, G)$. This contradicts that $M \subseteq \text{Streett}(R, G)$. Hence it follows that $\text{UniCloOmg}(\text{Streett}(R, G)) \subseteq \text{FinStreett}(R, G)$.
2. We now show the converse: $\text{UniCloOmg}(\text{Streett}(R, G)) \supseteq \text{FinStreett}(R, G)$. We have:

$$\begin{aligned} \text{FinStreett}(R, G) &= \{w \mid \limsup_k \text{dist}_k(w, (R, G)) < \infty\} \\ &= \bigcup_{B \in \mathbb{N}} \{w \mid \limsup_k \text{dist}_k(w, (R, G)) \leq B\} \\ &= \bigcup_{B \in \mathbb{N}} \bigcup_{n \in \mathbb{N}} \{w \mid \forall k \geq n, \text{dist}_k(w, (R, G)) \leq B\} \end{aligned}$$

The language $\{w \mid \forall k \geq n, \text{dist}_k(w, (R, G)) \leq B\}$ is closed, ω -regular, and included in $\text{Streett}(R, G)$. Hence $\text{FinStreett}(R, G) \subseteq \text{UniCloOmg}(\text{Streett}(R, G))$.

The result follows. ■

Lemma 2 naturally extends to finitary parity and finitary Büchi languages:

Corollary 1. The following assertions hold:

- For all $p : \Sigma \rightarrow \mathbb{N}$, we have $\text{UniCloOmg}(\text{Parity}(p)) = \text{FinParity}(p)$;
- For all $F \subseteq \Sigma$, we have $\text{UniCloOmg}(\text{Büchi}(F)) = \text{FinBüchi}(F)$.

Büchi languages are a special case of parity languages, and parity languages are in turn a special case of Streett languages. Since distance sequences for parity and Büchi languages have been defined as a special case of Streett languages, Corollary 1 follows from Lemma 2.

The following lemma states that finitary Büchi languages are Σ_2 -complete.

Theorem 3 (Topological characterization of finitary languages). *The finitary Büchi, finitary parity and finitary Streett are Σ_2 -complete.*

Proof. We show that if $\emptyset \subset F \subset \Sigma$, then $\text{FinBüchi}(F)$ is Σ_2 -complete. It follows from Corollary 1 that $\text{FinBüchi}(F) \in \Sigma_2$. We now show that $\text{FinBüchi}(F)$ is Σ_2 -hard. By Theorem 1 we have that $\text{Büchi}(F)$ is Π_2 -complete, hence $\Sigma^\omega \setminus \text{Büchi}(F)$ is Σ_2 -complete. We present a topological reduction to show that $\Sigma^\omega \setminus \text{Büchi}(F) \preceq \text{FinBüchi}(F)$. Let $b : \Sigma^\omega \rightarrow \Sigma^\omega$ be the stuttering function defined as follows:

$$\begin{aligned} w &= w_0 \ w_1 \ \dots \ w_n \ \dots \\ b(w) &= w_0 \underbrace{w_1 w_1}_2 \dots \underbrace{w_n w_n \dots w_n}_{2^n} \dots \end{aligned}$$

The function b is continuous. We check that the following holds:

$$\text{Inf}(w) \subseteq F \text{ iff } \exists B \in \mathbb{N}, \exists n \in \mathbb{N}, \forall k \geq n, \text{next}_k(b(w), F) \leq B.$$

Left to right direction: assume that from the position n of w , letters belong to F . Then from the position $2^n - 1$, letters of $b(w)$ belong to F , then $\text{next}_k(b(w), F) = 0$ for $k \geq 2^n - 1$.

Right to left direction: let B and n be integers such that for all $k \geq n$ we have $\text{next}_k(b(w), F) \leq B$. Assume $2^{k-1} > B$ and $k \geq n$, then the letter in position $2^k - 1$ in $b(w)$ is repeated 2^{k-1} times, thus $\text{next}_k(b(w), F)$ is either 0 or higher than 2^{k-1} . The latter is not possible since it must be less than B . It follows that the letter in position k in w belongs to F . Hence we get $\Sigma^\omega \setminus \text{Büchi}(F) \preceq \text{FinBüchi}(F)$, so $\text{FinBüchi}(F)$ is Σ_2 -complete. From this we deduce the two other claims as special cases. ■

4 Expressive power of finitary automata

In this section we consider the finitary automata, and compare their expressive power to classical automata. We then address the question of determinization. Deterministic finitary automata enjoy nice properties that allows to describe languages they recognize using the UniCloOmg operator. As a by-product we get boolean closure properties of finitary automata.

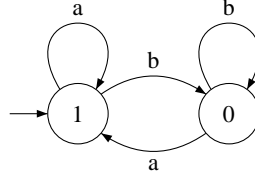


Fig. 1. A finitary Büchi automaton \mathcal{A}

4.1 Comparison with classical automata

Finitary conditions allow to express bounds requirements:

Example 1 ($DFB \not\subseteq \mathbb{L}_\omega$). Consider the finitary Büchi automaton shown in Fig. 1, the state labeled 0 being its only final state. Its language is $L_B = \{(b^{j_0} a^{f(0)}) \cdot (b^{j_1} a^{f(1)}) \cdot (b^{j_2} a^{f(2)}) \dots \mid f : \mathbb{N} \rightarrow \mathbb{N}, f \text{ bounded}, \forall i \in \mathbb{N}, j_i \in \mathbb{N}\}$. Indeed, 0-labeled state is visited while reading the letter b , and the 1-labeled state is visited while reading the letter a . An infinite word is accepted iff the 0-labeled state is visited infinitely often and there is a bound between two consecutive visits of the 0-labeled state. We can easily see that L_B is not ω -regular, using proof ideas from [BC06]: its complement would be ω -regular, so it would contain ultimately periodic words, which is not the case.

However, finitary automata cannot distinguish between “many b’s” and “only b’s”:

Example 2 ($DB \not\subseteq NFB$). Consider the language of infinitely many a ’s, *i.e.*, $L_I = \{w \mid w \text{ has an infinite number of } a\}$. The language L_I is recognized by a simple deterministic Büchi automaton. However, we can show that there is no finitary Büchi automata that recognizes L_I . Intuitively, such an automaton would, while reading the infinite word $w = ab \, ab^2 \, ab^3 \, ab^4 \dots ab^n \dots \in L_I$, have to distinguish between all b ’s, otherwise it would accept a word with only b ’s at the end. Assume towards contradiction that there exists \mathcal{A} a non-deterministic finitary Büchi automaton with N states recognizing L_I . Let us consider the infinite word w . Since w must be accepted by \mathcal{A} , there must be an accepting run ρ , represented as follows:

$$q_0 \xrightarrow{a} p_0 \xrightarrow{b} q_1 \dots q_n \xrightarrow{a} p_n \xrightarrow{b^{n+1}} q_{n+1} \dots$$

and

$$p_{n-1} \xrightarrow{b} q_{n,1} \xrightarrow{b} q_{n,2} \dots \xrightarrow{b} q_{n,n-1} \xrightarrow{b} q_{n,n} = q_n \dots$$

Since ρ is accepting, there exists $B \in \mathbb{N}$, and $n \in \mathbb{N}$, such that for all $k \geq n$ we have $\text{dist}_k(\rho, p) \leq B$. Let c be the lowest priority infinitely visited in ρ . As ρ is accepting, c is even. The state p_{k-1} is in position $\frac{k \cdot (k+1)}{2}$ in ρ . Let k be an

integer such that (a) $\frac{k \cdot (k+1)}{2} \geq n$ and (b) $k \geq (N+1) \cdot B$. Let us consider the set of states $\{q_{k,1}, \dots, q_{k,k}\}$. Since the distance function is bounded by B from the n -th position, the priority c appears at least once in each set of consecutively visited states of size B . Since $\frac{k \cdot (k+1)}{2} \geq n$ and $q_{k,1}$ is the state following p_{k-1} , the latter holds from $q_{k,1}$. Since $k \geq (N+1) \cdot B$, it appears at least $N+1$ times in $\{q_{k,1}, \dots, q_{k,k}\}$. Since there is N states in \mathcal{A} , at least one state has been reached twice. We can thus iterate: the infinite word $w' = ab \, ab^2 \, ab^3 \, ab^4 \dots b^{k-1} a \, b^\omega$, and the word w' is accepted by \mathcal{A} . However, $w' \notin L_I$ and hence we have a contradiction.

We summarize the results in the following theorem.

Theorem 4. *The following assertions hold: (a) $DB \not\subseteq NFB$; (b) $DFB \not\subseteq NB$.*

4.2 Deterministic finitary automata

Given a deterministic complete automaton \mathcal{A} with accepting condition Acc , we will consider the language obtained by using $\text{UniCloOmg}(Acc)$ as acceptance condition. Treating the automaton as a transducer, we consider the following function: $C_{\mathcal{A}} : \Sigma^\omega \rightarrow Q^\omega$ which maps an infinite word w to the unique run ρ of \mathcal{A} on w (there is a unique run since \mathcal{A} is deterministic and complete). Then:

$$\mathcal{L}(\mathcal{A}) = \{w \mid C_{\mathcal{A}}(w) \in Acc\} = C_{\mathcal{A}}^-(Acc).$$

We will focus on the following property: $C_{\mathcal{A}}^-(\text{UniCloOmg}(Acc)) = \text{UniCloOmg}(C_{\mathcal{A}}^-(Acc))$, which follows from the following lemma. Deterministic complete automata, regarded as transducers, preserve topology and ω -regularity. Hence applying the finitary operator UniCloOmg to the input (the language L) or to the acceptance condition Acc is equivalent.

Lemma 3. *For all $\mathcal{A} = (Q, \Sigma, q_0, \delta, Acc)$ deterministic complete automaton, we have:*

1. *for all $A \subseteq Q^\omega$, A is closed $\Rightarrow C_{\mathcal{A}}^-(A)$ closed ($C_{\mathcal{A}}$ is continuous).*
2. *for all $L \subseteq \Sigma^\omega$, L is closed $\Rightarrow C_{\mathcal{A}}(L)$ closed ($C_{\mathcal{A}}$ is closed).*
3. *for all $A \subseteq Q^\omega$, A is ω -regular $\Rightarrow C_{\mathcal{A}}^-(A)$ ω -regular.*
4. *for all $L \subseteq \Sigma^\omega$, L is ω -regular $\Rightarrow C_{\mathcal{A}}(L)$ ω -regular.*

Proof. We prove all the cases below.

1. Let $A \subseteq Q^\omega$ such that A is closed. Let w be such that for all $n \in \mathbb{N}$ we have $w_0 \dots w_n \in \text{pref}(C_{\mathcal{A}}^-(A))$. We define the run $\rho = C_{\mathcal{A}}(w)$ and show that $\rho = q_0 q_1 \dots \in A$. Since A is closed, we will show for all $n \in \mathbb{N}$

we have $q_0 \dots q_n \in \text{pref}(A)$. From the hypothesis we have $w_0 \dots w_{n-1} \in \text{pref}(C_{\mathcal{A}}^-(A))$, and then there exists an infinite word u such that $C_{\mathcal{A}}(w_0 \dots w_{n-1}u) \in A$. Let $C_{\mathcal{A}}(w_0 \dots w_{n-1}u) = q_0 q'_1 \dots q'_n \dots$, then we have $q_0 \xrightarrow{w_0} q'_1 \xrightarrow{w_1} q'_2 \dots \xrightarrow{w_{n-1}} q'_n \dots$. Since \mathcal{A} is deterministic, we get $q'_i = q_i$, and hence $q_0 \dots q_n \in \text{pref}(A)$.

2. Let $L \subseteq \Sigma^\omega$ such that L is closed. Let $\rho = q_0 q_1 \dots$ such that for all $n \in \mathbb{N}$ we have $q_0 \dots q_n \in \text{pref}(C_{\mathcal{A}}(L))$. Then for all $n \in \mathbb{N}$, there exists a word $w_0 w_1 \dots w_{n-1}$ such that $q_0 \xrightarrow{w_0} q_1 \xrightarrow{w_1} q_2 \dots \xrightarrow{w_{n-1}} q_n$, and $w_0 w_1 \dots w_{n-1} \in \text{pref}(L)$. We define by induction on n an infinite nested sequence of finite words $w_0 w_1 \dots w_n \in \text{pref}(L)$. We denote by w the limit of this nested sequence of finite words. We have that $\rho = C_{\mathcal{A}}(w)$. Since L is closed, $w \in L$.
3. Let $A \subseteq Q^\omega$ such that A recognized by a Büchi automaton $\mathcal{B} = (Q_{\mathcal{B}}, Q, P_0, \tau, F)$. We define the Büchi automaton $\mathcal{C} = (Q \times Q_{\mathcal{B}}, \Sigma, \{q_0\} \times P_0, \gamma, Q_{\mathcal{B}} \times F)$, where $(q_1, p_1) \xrightarrow{\sigma} (q_2, p_2)$ iff $q_1 \xrightarrow{\sigma} q_2$ in \mathcal{A} and $p_1 \xrightarrow{q_1} p_2$ in \mathcal{B} . We now show the correctness of our construction. Let $w = w_0 w_1 \dots$ accepted by \mathcal{C} , then there exists an accepting run ρ , as follows:

$$(q_0, p_0) \xrightarrow{w_0} (q_1, p_1) \xrightarrow{w_1} (q_2, p_2) \dots (q_n, p_n) \xrightarrow{w_n} (q_{n+1}, p_{n+1}) \dots$$

where the second component visits F infinitely often. Hence:

$$(\dagger) \begin{cases} q_0 \xrightarrow{w_0} q_1 \xrightarrow{w_1} q_2 \dots q_n \xrightarrow{w_n} q_{n+1} \dots \text{ in } \mathcal{A} \\ p_0 \xrightarrow{q_0} p_1 \xrightarrow{q_1} p_2 \dots p_n \xrightarrow{q_n} p_{n+1} \dots \text{ in } \mathcal{B} \end{cases}$$

Hence from (\dagger) , we have $C_{\mathcal{A}}(w) = q_0 q_1 \dots \in \mathcal{L}(\mathcal{B}) = A$, and it follows that $w \in C_{\mathcal{A}}^-(A)$. Conversely, let $w \in C_{\mathcal{A}}^-(A)$, then we have $\rho = C_{\mathcal{A}}(w) = q_0 q_1 \dots \in A = \mathcal{L}(\mathcal{B})$. Then the above statement (\dagger) holds, which entails that w is accepted by \mathcal{C} . It follows that \mathcal{C} recognizes $C_{\mathcal{A}}^-(A)$.

4. Let $L \subseteq \Sigma^\omega$ such that L is recognized by a Büchi automaton $\mathcal{B} = (Q_{\mathcal{B}}, \Sigma, P_0, \tau, F)$. We define the Büchi automaton $\mathcal{C} = (Q \times Q_{\mathcal{B}}, Q, \{q_0\} \times P_0, \gamma, Q \times F)$, where $(q, p_1) \xrightarrow{q} (q', p_2)$ iff there exists $\sigma \in \Sigma$, such that $q \xrightarrow{\sigma} q'$ in \mathcal{A} and $p_1 \xrightarrow{\sigma} p_2$ in \mathcal{B} . A proof similar to above show that \mathcal{C} recognizes $C_{\mathcal{A}}(L)$.

The desired result follows. ■

Theorem 5. *For any deterministic complete automaton $\mathcal{A} = (Q, \Sigma, q_0, \delta, \text{Acc})$ recognizing a language L , the finitary restriction of this automaton $\text{UniCloOmg}(\mathcal{A}) = (Q, \Sigma, q_0, \delta, \text{UniCloOmg}(\text{Acc}))$ recognizes $\text{UniCloOmg}(L)$.*

Proof. A word w is accepted by $\text{UniCloOmg}(\mathcal{A})$ iff $w \in C_{\mathcal{A}}^-(\text{UniCloOmg}(\text{Acc})) = \text{UniCloOmg}(C_{\mathcal{A}}^-(\text{Acc})) = \text{UniCloOmg}(L)$. ■

Theorem 5 allows to extend all known results on deterministic classes to finitary deterministic classes: as a corollary, we have $DFB \subset DFP$ and $DFP = DFS$.

We now show that non-deterministic finitary parity automata are more expressive than deterministic finitary parity automata. However, for every language $L \in \mathbb{L}_\omega$ there exists $\mathcal{A} \in DP$ such that \mathcal{A} recognizes L , and by Theorem 5 the deterministic finitary parity automaton $\text{UniCloOmg}(\mathcal{A})$ recognizes $\text{UniCloOmg}(L)$.

Corollary 2. *For every language $L \in \mathbb{L}_\omega$ there is a deterministic finitary parity automata \mathcal{A} such that $\mathcal{L}(\mathcal{A}) = \text{UniCloOmg}(L)$.*

Example 3 ($DFP \subset NFP$). As for Example 1 we consider the languages $L_1 = \{(a^{j_0}b^{f(0)}) \cdot (a^{j_1}b^{f(1)}) \cdot (a^{j_2}b^{f(2)}) \dots \mid f : \mathbb{N} \rightarrow \mathbb{N}, f \text{ bounded}, \forall i \in \mathbb{N}, j_i \in \mathbb{N}\}$ and $L_2 = \{(a^{f(0)}b^{j_0}) \cdot (a^{f(1)}b^{j_1}) \cdot (a^{f(2)}b^{j_2}) \dots \mid f : \mathbb{N} \rightarrow \mathbb{N}, f \text{ bounded}, \forall i \in \mathbb{N}, j_i \in \mathbb{N}\}$. It follows from Example 1 that both L_1 and L_2 belong to DFP , hence to NFP . A finitary parity automaton, relying on non-determinism, is easily built to recognize $L = L_1 \cup L_2$, hence $L \in NFP$. We can show that we cannot bypass this non-determinism, as by reading a word we have to decide well in advance which sequence will be bounded: a's or b's, i.e., $L \notin DFP$. To prove it, we interleave words of the form $(a^* \cdot b^*)^* \cdot a^\omega$ and $(a^* \cdot b^*)^* \cdot b^\omega$, and use a pumping argument to reach a contradiction. Assume towards contradiction that $L \in DFP$, and let \mathcal{A} be a deterministic complete finitary parity automaton with N states that recognizes L . Let q_0 be the starting state. Since a^ω belongs to L , its unique run on \mathcal{A} is accepting, and can be decomposed as follows: $q_0 \xrightarrow{a^{n_0}} s_0 \xrightarrow{a^{p_0}} s_0 \xrightarrow{a^{p_0}} \dots$ where s_0 is the lowest priority visited infinitely often while reading a^ω . Then, $a^{n_0}b^\omega$ belongs to this L , its unique run on \mathcal{A} is accepting, and has the following shape: $q_0 \xrightarrow{a^{n_0}} s_0 \xrightarrow{b^{n'_0}} t_0 \xrightarrow{b^{p'_0}} t_0 \xrightarrow{b^{p'_0}} \dots$ where t_0 is the lowest priority visited infinitely often while reading $a^{n_0}b^\omega$. Repeating this construction and by induction we have, as shown in Fig 2: where s_k is the low-

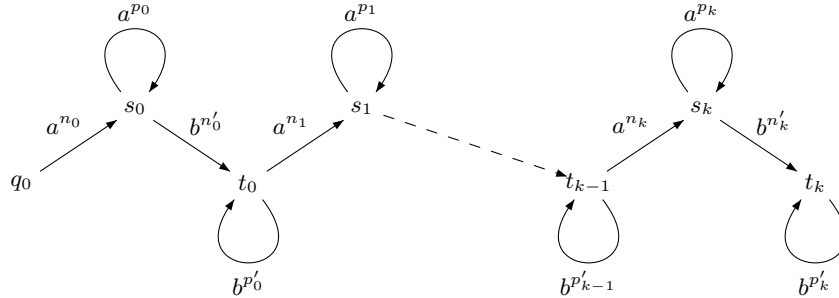


Fig. 2. Inductive construction showing that $L \notin DFP$.

est priority visited infinitely often while reading $a^{n_0}b^{n'_0} \dots a^{n_k}a^\omega$ and t_k is the lowest priority visited infinitely often while reading $a^{n_0}b^{n'_0} \dots a^{n_k}b^{n'_k}b^\omega$. There must be $i < j$, such that $t_i = t_j$. Let $u = a^{n_0}b^{n'_0} \dots b^{n'_i}$ and $v = b^{n'_{i+1}} \dots b^{n'_j}$, we have:

$$q_0 \xrightarrow{u} t_i \xrightarrow{a^{n_{i+1}}} s_{i+1} \xrightarrow{v} t_j = t_i$$

Consider the words $w = u \cdot (a^{n_{i+1}} \cdot v)^\omega$ and

$$w^* = u \cdot (b^{p'_i}a^{n_i+p_i}v) \cdot (b^{2p'_i}a^{n_i+2p_i}v) \dots (b^{kp'_i}a^{n_i+kp_i}v) \dots$$

w must be accepted by \mathcal{A} since it belongs to L . Hence w^* is accepted as well, but does not belong to L . We have a contradiction, and the result follows.

Theorem 6. *We have $DFP \subset NFP$.*

Observe that Theorem 5 does not hold for non-deterministic automata, since we have $DP = NP$ but $DFP \subset NFP$.

4.3 Non-deterministic finitary automata

We can show that non-deterministic finitary Streett automata can be reduced to non-deterministic finitary Büchi automata, and this would complete the picture of expressive power comparison. We first show that non-deterministic finitary Büchi automata are closed under intersection, and use it to show Theorem 7.

Lemma 4. *NFB is closed under intersection.*

Proof. Let $\mathcal{A}_1 = (Q_1, \Sigma, \delta_1, Q_0^1, F_1)$ and $\mathcal{A}_2 = (Q_2, \Sigma, \delta_2, Q_0^2, F_2)$ be two non-deterministic finitary Büchi automata. Without loss of generality we assume both \mathcal{A}_1 and \mathcal{A}_2 to be complete. We will define a construction similar to the synchronous product construction, where a switch between copies will happen while visiting F_1 or F_2 . The finitary Büchi automaton is $\mathcal{A} = (Q_1 \times Q_2 \times \{1, 2\}, \Sigma, \delta, Q_0^1 \times Q_0^2 \times \{1\}, F_1 \times Q_2 \times \{2\} \cup Q_1 \times F_2 \times \{1\})$. We define the transition relation δ below:

$$\begin{aligned} \delta = & \{((q_1, q_2, k), \sigma, (q'_1, q'_2, k)) \mid q'_1 \notin F_1, q'_2 \notin F_2, (q_1, \sigma, q'_1) \in \delta_1, (q_2, \sigma, q'_2) \in \delta_2, k \in \{1, 2\}\} \\ & \cup \{((q_1, q_2, 1), \sigma, (q'_1, q'_2, 2)) \mid q'_1 \in F_1, (q_1, \sigma, q'_1) \in \delta_1, (q_2, \sigma, q'_2) \in \delta_2\} \\ & \cup \{((q_1, q_2, 2), \sigma, (q'_1, q'_2, 1)) \mid q'_2 \in F_2, (q_1, \sigma, q'_1) \in \delta_1, (q_2, \sigma, q'_2) \in \delta_2\} \end{aligned}$$

Intuitively, the transition function δ is as follows: the first component mimics the transition for automata \mathcal{A}_1 , the second component mimics the transition for \mathcal{A}_2 , and there is a switch for the third component from 1 to 2 visiting a state in F_1 , and from 2 to 1 visiting a state in F_2 .

We now prove the correctness of the construction. Consider a word w that is accepted by \mathcal{A}_1 , and then there exists a bound B_1 and a run ρ_1 in \mathcal{A}_1 such that eventually, the number of steps between two visits to F_1 in ρ_1 is at most B_1 ; and similarly, there exists a bound B_2 and a run ρ_2 in \mathcal{A}_2 such that eventually the number of steps between two visits to F_2 in ρ_2 is at most B_2 . It follows that in our construction there is a run ρ (that mimics the runs ρ_1 and ρ_2) in \mathcal{A} such that eventually within $\max\{B_1, B_2\}$ steps a state in $F_1 \times Q_2 \times \{2\} \cup Q_1 \times F_2 \times \{1\}$ is visited in ρ . Hence w is accepted by \mathcal{A} . Conversely, consider a word w that is accepted by \mathcal{A} , and let ρ be a run and B be the bound such that eventually between two visits to the accepting states in ρ is separated by at most B steps. Let ρ_1 and ρ_2 be the decomposition of the run ρ in \mathcal{A}_1 and \mathcal{A}_2 , respectively. It follows that both in \mathcal{A}_1 and \mathcal{A}_2 the respective final states are eventually visited within at most $2 \cdot B$ steps in ρ_1 and ρ_2 , respectively. It follows that w is accepted by both \mathcal{A}_1 and \mathcal{A}_2 . Hence we have $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}_1) \cap \mathcal{L}(\mathcal{A}_2)$. ■

Theorem 7. *We have $NFB = NFP = NFS$.*

Proof. We will present a reduction of NFS to NFB and the result will follow. Since the Streett condition is a finite conjunction of conditions $\text{Inf}(w) \cap R_i \neq \emptyset \Rightarrow \text{Inf}(w) \cap G_i \neq \emptyset$, by Lemma 4 it suffices to handle the special case when $d = 1$. Hence we consider a non-deterministic Streett automaton $\mathcal{A} = (Q, \Sigma, \delta, Q_0, (R, G))$ with $(R, G) = (R_1, G_1)$. Without loss of generality we assume \mathcal{A} to be complete. We construct a non-deterministic Büchi automaton $\mathcal{A}' = (Q \times \{1, 2, 3\}, \Sigma, \delta', Q_0 \times \{1\}, Q \times \{2\})$, where the transition relation δ' is given as follows:

$$\begin{aligned} \delta' = & \{(q, 1), \sigma, (q', j) \mid (q, \sigma, q') \in \delta, j \in \{1, 2\}\} \\ & \cup \{(q, 2), \sigma, (q', 2) \mid q' \notin R_1, (q, \sigma, q') \in \delta\} \\ & \cup \{(q, 2), \sigma, (q', 3) \mid q' \in R_1, (q, \sigma, q') \in \delta\} \\ & \cup \{(q, 3), \sigma, (q', 3) \mid q' \notin G_1, (q, \sigma, q') \in \delta\} \\ & \cup \{(q, 3), \sigma, (q', 2) \mid q' \in G_1, (q, \sigma, q') \in \delta\} \end{aligned}$$

In other words, the state component mimics the transition of \mathcal{A} , and in the second component: (a) the automaton can choose to stay in component 1, or switch to 2; (b) there is a switch from 2 to 3 upon visiting a state in R_1 ; and (c) there is a switch from 3 to 2 upon visiting a state in G_1 . Consider a word w accepted by \mathcal{A} and an accepting run ρ in \mathcal{A} , and let B be the bound on the distance sequence. We show that w is accepted by \mathcal{A}' by constructing an accepting run ρ' in \mathcal{A}' . We consider the following cases:

1. If infinitely many requests R_1 are visited in ρ , then in \mathcal{A}' immediately switch to component 2, and then mimic the run ρ as a run ρ' in \mathcal{A}' . It follows that

from some point j on every request is granted within B steps, and it follows that after position j , whenever the second component is 3, it becomes 2 within B steps. Hence w is accepted by \mathcal{A} .

2. If finitely many requests R_1 are visited in ρ , then after some point j , there are no more requests. The automaton \mathcal{A}' mimics the run ρ by staying in the second component as 1 for j steps, and then switches to component 2. Then after j steps we always have the second component as 2, and hence the word is accepted.

Conversely, consider a word w accepted by \mathcal{A}' and consider the accepting run ρ' . We mimic the run in \mathcal{A} . To accept the word w , the run ρ' must switch to the second component as 2, say after j steps. Then, from some point on whenever a state with second component 3 is visited, within some bound B steps a state with second component 2 is visited. Hence the run ρ is accepting in \mathcal{A} . Thus the languages of \mathcal{A} and \mathcal{A}' coincide, and the desired result follows. ■

Our results are summarized in Corollary 3 and shown in Fig 3.

Corollary 3. *We have (a) $DFB \not\subseteq \mathbb{L}_\omega$; (b) $DFB \subset DFP = DFS \subset NFB = NFP = NFS$; (c) $DB \not\subseteq NFB$; (d) $\mathbb{L}_\omega \not\subseteq NFB$.*

4.4 Closure properties

Theorem 8 (Closure properties). *The following closure properties hold:*

1. *DFP is closed under intersection.*
2. *DFP is not closed under union.*
3. *NFP is closed under union and intersection.*
4. *DFP and NFP are not closed under complementation.*

Proof. We prove all the cases below.

1. Intersection closure for DFP follows from Theorem 5 and from the observation that for all $L, L' \subseteq \Sigma^\omega$ we have $\text{UniCloOmg}(L \cap L') = \text{UniCloOmg}(L) \cap \text{UniCloOmg}(L')$. The observation is proved as follows. Let $M \in \Pi_1 \cap \mathbb{L}_\omega$ and $M \subseteq L \cap L'$, then $M \subseteq \text{UniCloOmg}(L) \cap \text{UniCloOmg}(L')$, and hence $\text{UniCloOmg}(L \cap L') \subseteq \text{UniCloOmg}(L) \cap \text{UniCloOmg}(L')$. Conversely, let $M_1 \subseteq \text{UniCloOmg}(L)$ and $M_2 \subseteq \text{UniCloOmg}(L')$, then $M_1 \cap M_2 \in \Pi_1 \cap \mathbb{L}_\omega$ and $M_1 \cap M_2 \subseteq L \cap L'$. Hence $M_1 \cap M_2 \subseteq \text{UniCloOmg}(L \cap L')$, thus $\text{UniCloOmg}(L) \cap \text{UniCloOmg}(L') \subseteq \text{UniCloOmg}(L \cap L')$.
2. Failure of closure under union for DFP follows from Example 3.
3. Union closure for NFP is easy and relies on non-determinism, while intersection closure follows from Lemma 4, since $NFP = NFB$.

4. Failure of closure under complementation for DFP follows from items 1. and 2., since this closure together with intersection closure would imply union closure. Failure of closure under complementation for NFP follows from Example 2. Indeed, the language $L_F = \{a, b\}^\omega \setminus L_I = \{w \mid w \text{ has a finite number of } a\}$ lies in NFP ; however, Example 2 shows that its complement is not expressible by non-deterministic finitary Büchi automata, hence nor by non-deterministic finitary parity automata.

The result follows. ■

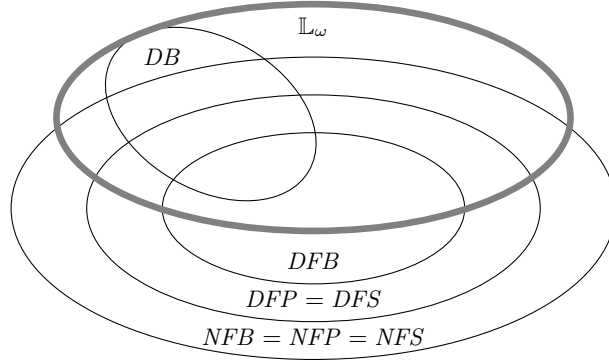


Fig. 3. Expressive power classification

5 Regular Expression Characterization

In this section we address the question of giving a syntactical representation of finitary languages, using a special class of regular expressions.

The class of ωB -regular expressions was introduced in the work of [BC06] as an extension of ω -regular expressions, as an attempt to express bounds in regular languages. To define ωB -regular expressions, we need regular expressions and ω -regular expressions.

Regular expressions define regular languages over finite words, and have the following grammar:

$$L := \emptyset \mid \varepsilon \mid \sigma \mid L \cdot L \mid L^* \mid L + L; \quad \sigma \in \Sigma$$

In the above grammar, \cdot stands for concatenation, $*$ for Kleene star and $+$ for union. Then ω -regular languages are finite union of $L \cdot L'^\omega$, where L and L' are regular languages of finite words. The class of ωB -regular languages, as defined

in [BC06], is described by finite union of $L \cdot M^\omega$, where L is a regular language over finite words and M is a B -regular language over infinite sequences of finite words. The grammar for B -regular languages is as follows:

$$M := \emptyset \mid \varepsilon \mid \sigma \mid M \cdot M \mid M^* \mid M^B \mid M + M; \quad \sigma \in \Sigma$$

The semantics of regular languages over infinite sequences of finite words will assign to a B -regular expression M , a language in $(\Sigma^*)^\omega$. The infinite sequence $\langle u_0, u_1, \dots \rangle$ will be denoted by \mathbf{u} . The semantics is defined by structural induction as follows.

- \emptyset is the empty language,
- ε is the language containing the single sequence $(\varepsilon, \varepsilon, \dots)$,
- a is the language containing the single sequence (a, a, \dots) ,
- $M_1 \cdot M_2$ is the language $\{\langle u_0 \cdot v_0, u_1 \cdot v_1, \dots \rangle \mid \mathbf{u} \in M_1, \mathbf{v} \in M_2\}$,
- M^* is the language $\{\langle u_1 \dots u_{f(1)-1}, u_{f(1)} \dots u_{f(2)-1}, \dots \rangle \mid \mathbf{u} \in M, f : \mathbb{N} \rightarrow \mathbb{N}\}$,
- M^B is defined like M^* but we additionally require the values $f(i+1) - f(i)$ to be bounded uniformly in i ,
- $M_1 + M_2$ is $\{\mathbf{w} \mid \mathbf{u} \in M_1, \mathbf{v} \in M_2, \forall i, w_i \in \{u_i, v_i\}\}$.

Finally, the ω -operator on sequences with nonempty words on infinitely many coordinates is: $\langle u_0, u_1, \dots \rangle^\omega = u_0 u_1 \dots$. This operation is naturally extended to languages of sequences by taking the ω power of every sequence in the language. The class of ωB -regular languages is more expressive than NFB , and this is due to the $*$ -operator. We will consider the following fragment of ωB -regular languages where we do not use the $*$ -operator for B -regular expressions (however, the $*$ -operator is allowed for L , regular languages over finite words). We call this fragment the star-free fragment of ωB -regular languages. In the following two lemmas we show that star-free ωB -regular expressions express exactly NFB .

Lemma 5. *All languages in NFB can be described by a star-free ωB -regular expression.*

Proof. Let $\mathcal{A} = (Q, \Sigma, \delta, Q_0, F)$ be a non-deterministic finitary Büchi automaton. Without loss of generality we assume $Q = \{1, \dots, n\}$. Let $L_{q,q'} = \{u \in \Sigma^* \mid q \xrightarrow{u} q'\}$ and $M_q^{\geq c} = \{\mathbf{u} \mid (|u_i|)_i \text{ is bounded and } \forall i, q \xrightarrow{u_i} q\}$. Then

$$\mathcal{L}(\mathcal{A}) = \bigcup_{q_0 \in Q_0, q \in F} L_{q_0, q} \cdot (M_q)^\omega.$$

For all $q, q' \in Q$ we have $L_{q,q'} \subseteq \Sigma^*$ is regular. We now show that for all $q \in Q$ the language M_q is B -regular. For all $0 \leq k \leq n$ and $q, q' \in Q$, let $M_{q,q'}^k = \{\mathbf{u} \mid$

$(|u_i|)_i$ is bounded and $\forall i, q \xrightarrow{u_i} q'$ where all intermediate visited states are from $\{1, \dots, k\}$. We show by induction on $0 \leq k \leq n$ that for all $q, q' \in Q$ the language $M_{q,q'}^k$ is B -regular. The base case $k = 0$ follows from observation:

$$M_{q,q'}^0 = \begin{cases} a_1 + a_2 + \dots + a_l & \text{if } q \neq q' \text{ and } (q, a, q') \in \delta \iff \exists i \in \{1, \dots, l\}, a = a_i \\ \varepsilon + a_1 + a_2 + \dots + a_l & \text{if } q = q' \text{ and } (q, a, q') \in \delta \iff \exists i \in \{1, \dots, l\}, a = a_i \\ \emptyset & \text{otherwise} \end{cases}$$

The inductive case for $k > 0$ follows from observation:

$$M_{q,q'}^k = M_{q,k}^{k-1} \cdot (M_{k,k}^{k-1})^B \cdot M_{k,q'}^{k-1} + M_{q,q'}^{k-1}$$

Since $M_{q,q}^n = M_q$, we conclude that $\mathcal{L}(\mathcal{A})$ is described by a star-free ωB -regular expression. \blacksquare

Lemma 6. *All languages described by a star-free ωB -regular expression is recognized by a non-deterministic finitary Büchi automaton.*

Proof. To prove this result, we will describe automata reading infinite sequences of finite words, and corresponding acceptance conditions. Let $\mathcal{A} = (Q, \Sigma, \delta, Q_0, F)$ a finitary Büchi automaton. While reading an infinite sequence u of finite words, \mathcal{A} will accept if the following conditions are satisfied: (1) $\exists q_0 \in Q_0, q_1, q_2, \dots \in F, \forall i \in \mathbb{N}$, we have $q_i \xrightarrow{u_i} q_{i+1}$ and (2) $(|u_n|)_n$ is bounded.

We show that for all M star-free B -regular expression, there exists a non-deterministic finitary Büchi automaton accepting M^B , language of infinite sequence of finite words, as described above. We proceed by induction on M .

- The cases \emptyset, ε and $a \in \Sigma$ are easy.
- From M to M^B , the same automaton for M works for M^B as well, since B is idempotent.
- From M_1, M_2 to $M_1 + M_2$: this involves non-determinism. The automaton guesses for each finite word which word is used. Let $\mathcal{A}_1 = (Q_1, \Sigma, \delta_1, Q_1^0, F_1)$ and $\mathcal{A}_2 = (Q_2, \Sigma, \delta_2, Q_2^0, F_2)$ two non-deterministic finitary Büchi automata accepting M_1^B and M_2^B , respectively. For $k \in \{1, 2\}$ and $T \subseteq Q_k$, we define $\text{Final}(T) = \{q' \in F_k \mid \exists q \in T, \exists u \in \Sigma^*, q \xrightarrow{u}_{\mathcal{A}_k} q'\}$ to be the state of final states reachable from a state in T . We denote by Final^k the k -th iteration of Final , e.g., $\text{Final}^3(T) = \text{Final}(\text{Final}(\text{Final}(T)))$.

We define a finitary Büchi automaton:

$$\mathcal{A} = (\underbrace{(Q_1 \times 2^{Q_1}) \cup (Q_2 \times 2^{Q_1})}_{\text{computation states}} \cup \underbrace{2^{Q_1} \times 2^{Q_2}}_{\text{guess states}}, \Sigma, \delta, (Q_1^0, Q_2^0), F)$$

where

$$\begin{aligned}
\delta = & \{((Q, Q'), \varepsilon, (q, \text{Final}(Q'))) \mid q \in Q\} && \text{(guess is 1)} \\
& \cup \{((Q, Q'), \varepsilon, (q', \text{Final}(Q))) \mid q' \in Q'\} && \text{(guess is 2)} \\
& \cup \{((q, T), \sigma, (q', T)) \mid (q, \sigma, q') \in \delta_1 \cup \delta_2\} \\
& \cup \{((q_1, T), \varepsilon, (\{q_1\}, T)) \mid q_1 \in F_1\} \\
& \cup \{((q_2, T), \varepsilon, (T, \{q_2\})) \mid q_2 \in F_2\}
\end{aligned}$$

There are two kinds of states. Computation states are (q, T) where $q \in Q_1$ and $T \subseteq Q_2$ (or symmetrically $q \in Q_2$ and $T \subseteq Q_1$), where q is the current state of the automaton that has been decided to use for the current finite word, and T is the set of final states of the other automaton that would have been reachable if one had chosen this automaton. Guess states are (Q, Q') , where Q is the set of states from \mathcal{A}_1 one can start reading the next word, and similarly for Q' .

We now prove the correctness of our construction. Consider an infinite sequence w accepted by \mathcal{A} , and consider an accepting run ρ . There are three cases:

1. either all guesses are 1;
2. or all guesses are 2;
3. else, both guesses happen.

The first two cases are symmetric. In the first, we can easily see that w is accepted by \mathcal{A}_1 , and similarly in the second w is accepted by \mathcal{A}_2 .

We now consider the third case. There are two symmetric subcases: either the first guess is 1, then

$$\rho = (Q_1^0, Q_2^0) \cdot (q_1^0, \text{Final}(Q_2^0)) \dots,$$

with $q_1^0 \in Q_1^0$; or symmetrically the first guess is 2, then

$$\rho = (Q_1^0, Q_2^0) \cdot (q_2^0, \text{Final}(Q_1^0)) \dots,$$

with $q_2^0 \in Q_2^0$. We consider only the first subcase. Then

$$\rho = (Q_1^0, Q_2^0) \cdot (q_1^0, \text{Final}(Q_2^0)) \dots (q_1^1, \text{Final}(Q_2^0)) \cdot (\{q_1^1\}, \text{Final}(Q_2^0)) \dots,$$

where u_0 is a finite prefix of w^ω such that $q_1^0 \xrightarrow{u_0} q_1^1$ in \mathcal{A}_1 and $q_1^1 \in F_1$. We denote by ρ_0 the finite prefix of ρ up to $(q_1^1, \text{Final}(Q_2^0))$. Let k be the first time when guess is 2: then

$$\rho = \rho_0 \cdot \rho_1 \cdot \rho_{k-1} \cdot (\{q_1^k\}, \text{Final}^k(Q_2^0)) \cdot (q_2^0, \text{Final}(\{q_k\})) \dots,$$

where $q_2^0 \in \text{Final}^k(Q_2^0)$ and for $1 \leq i \leq k-1$, we have

$$\rho_i = (\{q_1^i\}, \text{Final}^i(Q_2^0)) \cdot (q_1^i, \text{Final}^{i+1}(Q_2^0)) \dots (q_1^{i+1}, \text{Final}^{i+1}(Q_2^0)),$$

and u_i is a finite word such that $q_1^i \xrightarrow{u_i} q_1^{i+1}$ in \mathcal{A}_1 , $q_1^{i+1} \in F_1$ and $u_0 u_1 \dots u_{k-1}$ finite prefix of w^ω . Since $q_2^0 \in \text{Final}^k(Q_2^0)$, there exists v_0, v_1, \dots, v_{k-1} finite words and $q_2^1, \dots, q_2^k \in F_2$ such that: $q_2^0 \xrightarrow{v_0} q_2^1 \xrightarrow{v_1} \dots \xrightarrow{v_{k-1}} q_2^k$. Then we can repeat this by induction, constructing $u \in M_1^B$ and $v \in M_2^B$, such that for all $i \in \mathbb{N}$, we have $w_i \in \{u_i, v_i\}$.

Conversely, let $u \in M_1^B$ and $v \in M_2^B$, and w such that $\forall i \in \mathbb{N}, w_i \in \{u_i, v_i\}$. Using \mathcal{A}_1 when $w_i = u_i$ and \mathcal{A}_2 otherwise, one can construct an accepting run for w and \mathcal{A} . Hence \mathcal{A} recognizes $(M_1 + M_2)^B$.

- From M_1, M_2 to $M_1 \cdot M_2$: the automaton keeps tracks of pending states while reading the other word. Let $\mathcal{A}_1 = (Q_1, \Sigma, \delta_1, Q_1^0, F_1)$ and $\mathcal{A}_2 = (Q_2, \Sigma, \delta_2, Q_2^0, F_2)$ two non-deterministic finitary Büchi automata accepting M_1^B and M_2^B , respectively. Let $\mathcal{A} = ((Q_1 \times F_2) \cup (Q_2 \times F_1), \Sigma, \delta, Q_1^0 \times Q_2^0, F_1 \times F_2)$, where

$$\begin{aligned} \delta = & \{((q, f), \sigma, (q', f)) \mid (q, \sigma, q') \in \delta_1, f \in F_2\} \\ & \cup \{((q, f), \sigma, (q', f)) \mid (q, \sigma, q') \in \delta_2, f \in F_1\} \\ & \cup \{((q_1, f), \varepsilon, (f, q_1)) \mid q_1 \in F_1\} \\ & \cup \{((q_2, f), \varepsilon, (f, q_2)) \mid q_2 \in F_2\} \end{aligned}$$

Intuitively, the transition relation is as follows: either one is reading using \mathcal{A}_1 or \mathcal{A}_2 . In both cases, the automaton remembers the last final state visited while reading in the other automaton in order to restore this state for the next word. Let w accepted by \mathcal{A} , an accepting run is as follows:

$$(q_1^0, q_2^0) \xrightarrow{w_0} (q_1^1, q_2^1) \xrightarrow{w_1} \dots (q_1^i, q_2^i) \xrightarrow{w_i} (q_1^{i+1}, q_2^{i+1}) \dots$$

where $(q_1^0, q_2^0) \in Q_1^0 \times Q_2^0$, for all $i \geq 1$, we have $(q_1^i, q_2^i) \in F_1 \times F_2$ and $(|w_n|)_n$ bounded. From the construction, for all $i \in \mathbb{N}$, we have $w_i = u_i^0 \cdot v_i^0 \cdot u_i^1 \cdot v_i^1 \dots u_i^{k_i} \cdot v_i^{k_i}$, where

$$q_1^i = q_1^i(0) \xrightarrow{u_i^0} q_1^i(1) \xrightarrow{u_i^1} q_1^i(2) \dots \xrightarrow{u_i^{k_i}} q_1^i(k_i + 1) = q_1^{i+1} \quad \text{in } \mathcal{A}_1$$

$$q_2^i = q_2^i(0) \xrightarrow{v_i^0} q_2^i(1) \xrightarrow{v_i^1} q_2^i(2) \dots \xrightarrow{v_i^{k_i}} q_2^i(k_i + 1) = q_2^{i+1} \quad \text{in } \mathcal{A}_2$$

the states $(q_1^i(k), q_2^i(k))$ belong to $F_1 \times F_2$. We define $u_i = u_i^0 u_i^1 \dots u_i^{k_i}$ and $v_i = v_i^0 v_i^1 \dots v_i^{k_i}$. From the above follows that u and v are accepted by \mathcal{A}_1 and \mathcal{A}_2 , respectively. Then $w \in (M_1 \cdot M_2)^B$.

Conversely, a sequence in $(M_1 \cdot M_2)^B$ is clearly accepted by \mathcal{A} . Hence \mathcal{A} recognizes $(M_1 \cdot M_2)^B$.

We now prove that all star-free ωB -regular expressions are recognized by a non-deterministic finitary Büchi automaton. Since NFB are closed under finite

union (Theorem 8), we only need to consider expressions $L \cdot M^\omega$, where $L \subseteq \Sigma^*$ is regular language of finite words and M star-free B -regular expression. The constructions above ensure that there exists $\mathcal{A}_M = (Q_M, \Sigma, \delta_M, Q_M^0, F_M)$, a non-deterministic finitary Büchi automaton that recognizes the language M^B of infinite sequences. Let $\mathcal{A}_L = (Q_L, \Sigma, \delta_L, Q_L^0, F_L)$ be a finite automaton over finite words that recognizes L . We construct a non-deterministic finitary Büchi automaton as follows: $\mathcal{A} = (Q_L \cup Q_M, \Sigma, \delta, Q_L^0, F_M)$ where $\delta = \delta_L \cup \delta_M \cup \{(q, \varepsilon, q') \mid q \in F_L, q' \in Q_M^0\}$. In other words, first \mathcal{A} simulates \mathcal{A}_L , and when a finite prefix is recognized by \mathcal{A}_L , then \mathcal{A} turns to \mathcal{A}_M and simulates it.

We argue that \mathcal{A} recognizes $L \cdot M^\omega$. Let w accepted by \mathcal{A} , and u the finite prefix read by \mathcal{A}_L , $w = u \cdot v$. From v infinite word, we define \mathbf{v} an infinite sequence of finite words by sequencing v each time a final state (i.e., from F_L) is visited. The sequence \mathbf{v} is accepted by \mathcal{A}_M , hence belongs to M^B , and since $\mathbf{v}^\omega = v$, we have $v \in (M^B)^\omega = M^\omega$, and finally $w \in L \cdot M^\omega$. Conversely, let $w = u \cdot \mathbf{v}^\omega$, where $u \in L$ and $\mathbf{v} \in M^B$. Let $q_0 \in Q_L^0, q \in F_L$ such that $q_0 \xrightarrow{u} q$. Let $q' \in Q_0, q_1, q_2, \dots \in F_L$, such that for all $i \in \mathbb{N}$ we have $q_i \xrightarrow{v_i} q_{i+1}$. The key, yet simple observation is that for all star-free B -regular expressions M and for all $\mathbf{v} \in M$ we have $(|v_n|)_n$ is bounded. This is straightforward by induction on M . Hence, from position $|u|$, the set F_L is visited infinitely many times, and there is a bound between two consecutive visits. Thus w is accepted by \mathcal{A} . ■

The following theorem follows from Lemma 5 and Lemma 6.

Theorem 9. *NFB has exactly the same expressive power as star-free ωB -regular expressions.*

6 Decision Problems

In this section we consider the complexity of the decision problems for finitary languages. We present the results for finitary Büchi automata for simplicity, but the arguments for finitary parity and Streett automata are similar.

For the proofs of the results of this section we need to consider co-Büchi conditions (dual of Büchi conditions): given a set F , it requires that elements that appear infinitely often are outside F , in other words, elements in F appear only finitely often. It maybe noted that co-Büchi and finitary co-Büchi conditions coincide. We will also consider co-finitary Büchi condition, that is the complement of a finitary Büchi condition: given a set F co-finitary Büchi condition for F is the complement of $\text{FinBüchi}(F)$, that is $\Sigma^\omega \setminus \text{FinBüchi}(F)$.

Lemma 7. *Let $\mathcal{A} = (Q, \Sigma, Q_0, \delta, F_b, F_c)$ be an automaton with F_b and F_c are subsets of Q . Consider the acceptance condition Φ_1 as the conjunction of the*

finitary Büchi condition with set F_b , and the co-finitary Büchi condition with set F_c ; and the acceptance condition Φ_2 as the conjunction of Büchi condition with set F_b , and the co-Büchi condition with set F_c . The following assertions hold:

1. The answer of the emptiness problem of \mathcal{A} for Φ_2 is Yes iff there is a cycle C in \mathcal{A} such that $C \cap F_b \neq \emptyset$ and $C \cap F_c = \emptyset$.
2. The answer of the emptiness problem for Φ_1 and Φ_2 coincide.
3. The emptiness problem for Φ_1 is decidable in NLOGSPACE.

Proof. We prove the results as follows.

1. We first prove parts 1. and 2. Without loss of generality we assume that for all $q \in Q$, there exists a path from an initial state $q_0 \in Q$ to q (otherwise we can delete q). If there is a cycle C with $C \cap F_b \neq \emptyset$ and $C \cap F_c = \emptyset$, then consider a finite word u to reach C , and a word v that execute C . The word $u \cdot v^\omega$ is a witness that \mathcal{A} with Φ_1 as well as Φ_2 is non-empty. Conversely, the condition Φ_2 is a Rabin 1-pair condition, and by existence of memoryless strategies for Rabin condition [EJ88], it follows that if \mathcal{A} is non-empty for Φ_2 , then there must be a cycle C in \mathcal{A} such that $C \cap F_b \neq \emptyset$ and $C \cap F_c = \emptyset$. The condition Φ_1 can be specified as a finitary parity condition with three priorities (1, 2, 3) by assigning priority 1 to states in F_c , 2 to states in $F_b \setminus F_c$, and 3 to the rest. By existence of memoryless strategies for finitary parity objectives [CHH09], it follows that if \mathcal{A} is non-empty for Φ_1 , then there must be a cycle C in \mathcal{A} such that $C \cap F_b \neq \emptyset$ and $C \cap F_c = \emptyset$. The result follows.
2. The result follows from the emptiness problem of non-deterministic Rabin 1-pair automata. The basic idea of the proof is as follows: we show that the witness cycle C can be guessed and verified in logarithmic space. The guesses are as follows: (a) first the initial prefix of the path to C is guessed by guessing one state (the next state) at a time (hence only one guess is made at a time which is logarithmic space), (b) then the starting state of the cycle C is guessed and stored (again in logarithmic space), and (c) the cycle is guessed by again considering one state at a time and at each step it is verified that the state generated is in $Q \setminus F_c$; (d) one state in the cycle such that the state is in F_b is guessed and verified; and (e) finally it is checked that the cycle is completed by visiting the starting state of the cycle. Hence at every step only constantly many guesses are made, stored and verified. The NLOGSPACE upper bound follows.

The desired result follows. ■

Theorem 10 (Decision problems). *The following assertions hold:*

1. (Emptiness). Given a finitary Büchi automaton \mathcal{A} , whether $\mathcal{L}(\mathcal{A}) = \emptyset$ is NLOGSPACE-complete and can be decided in linear time.
2. (Universality). Given a finitary Büchi automaton \mathcal{A} whether $\mathcal{L}(\mathcal{A}) = \Sigma^\omega$ is PSPACE-complete.
3. (Language inclusion). Given two finitary Büchi automata \mathcal{A} and \mathcal{B} , whether $\mathcal{L}(\mathcal{A}) \subseteq \mathcal{L}(\mathcal{B})$ is PSPACE-complete.

Proof. We show the three parts of the proof.

1. The NLOGSPACE upper bound follows from Lemma 7: we consider the special case where the set F_c is empty. The NLOGSPACE lower bound follows from NLOGSPACE-hardness of reachability problem in a directed graph: given s and t two vertices, is there a path from s to t ? Given a directed graph and s, t two vertices, the corresponding automaton has s as initial vertex, t as unique final vertex, and we add a self-loop over t . Then there is a path from s to t if and only if the language accepted by this finitary Büchi automaton is non-empty. This concludes since $\text{co-NLOGSPACE} = \text{NLOGSPACE}$.
2. The PSPACE upper bound will follow from the following PSPACE upper bound for language inclusion, item 3. The PSPACE lower bound follows from the PSPACE lower bound for finite automata. The universality problem for automata over finite words is PSPACE-hard even when all the accepting states are absorbing [MS72]. For such automata over finite words the acceptance is the same as for finitary Büchi condition. The result follows.
3. The PSPACE lower bound follows from item 2. by the PSPACE-hardness for universality. We now present the PSPACE upper bound. Let $\mathcal{A} = (Q_A, \Sigma, Q_{A,0}, \delta_A, F_A)$ and $\mathcal{B} = (Q_B, \Sigma, Q_{B,0}, \delta_B, F_B)$ be two finitary Büchi automata. Let $\mathcal{A} \times \overline{\mathcal{B}} = (Q_A \times 2^{Q_B}, \Sigma, (Q_{A,0}, Q_{B,0}), \delta, F_b, F_c)$ be an automaton where for all $s \in Q_A, S \subseteq Q_B$ and $\sigma \in \Sigma$,

$$\delta((s, S), \sigma) = \bigcup_{q \in S} \{(s', q') \mid s' \in \delta_A(s, \sigma), q' \in \delta_B(q, \sigma)\}$$

and $F_b = \{(s, S) \mid s \in F_A\}$ and $F_c = \{(s, S) \mid S \cap F_B = \emptyset\}$. In other words $\mathcal{A} \times \overline{\mathcal{B}}$ is synchronous product of \mathcal{A} and the power set (subset construction) of \mathcal{B} . The acceptance condition is the conjunction of the finitary Büchi condition with set F_b and co-finitary Büchi condition with set F_c .

We claim that $\mathcal{L}(\mathcal{A} \times \overline{\mathcal{B}}) = \emptyset$ iff $\mathcal{L}(\mathcal{A}) \subseteq \mathcal{L}(\mathcal{B})$.

Assume $\mathcal{L}(\mathcal{A} \times \overline{\mathcal{B}}) \neq \emptyset$, then there is a cycle C such that $C \cap F_b \neq \emptyset$ and $C \cap F_c = \emptyset$. The lasso word that executes the finite path to reach C and then execute it forever is a witness word that is accepted by \mathcal{A} but not by \mathcal{B} . Hence $\mathcal{L}(\mathcal{A}) \not\subseteq \mathcal{L}(\mathcal{B})$.

Assume $\mathcal{L}(\mathcal{A} \times \overline{\mathcal{B}}) = \emptyset$, then every words accepted by \mathcal{A} is not accepted by $\overline{\mathcal{B}}$, hence accepted by \mathcal{B} . Thus $\mathcal{L}(\mathcal{A}) \subseteq \mathcal{L}(\mathcal{B})$.

Since the construction is exponential and the non-emptiness problem can be decided in NLOGSPACE (Lemma 7), we obtain a $\text{NPSPACE} = \text{PSPACE}$ upper bound.

The result follows. ■

Acknowledgements. We thank Thomas Colcombet for explaining to us results related to ωB -automata.

References

- AH98. Rajeev Alur and Thomas A. Henzinger. Finitary Fairness. *ACM Transactions on Programming Languages and Systems*, 20(6):1171–1194, 1998.
- AS85. Bowen Alpern and Fred B. Schneider. Defining Liveness. *Information Processing Letters*, 21(4):181–185, 1985.
- BC06. Mikolaj Bojanczyk and Thomas Colcombet. Bounds in ω -Regularity. In *Proceedings of the 21st Annual IEEE Symposium on Logic in Computer Science, LICS'06*, pages 285–296. IEEE Computer Society, 2006.
- Boj10. Mikolaj Bojanczyk. Beyond omega-regular languages. In *International Symposium on Theoretical Aspects of Computer Science, STACS'*, pages 11–16, 2010.
- BT09. Mikolaj Bojanczyk and Szymon Torunczyk. Deterministic automata and extensions of weak mso. In *International Conference on the Foundations of Software Technology and Theoretical Computer Science, FSTTCS'*, pages 73–84, 2009.
- Büc62. J. Richard Büchi. On a decision method in restricted second-order arithmetic. In *Proceedings of the 1st International Congress of Logic, Methodology, and Philosophy of Science, CLMPS'60*, pages 1–11. Stanford University Press, 1962.
- CHH09. Krishnendu Chatterjee, Thomas A. Henzinger, and Florian Horn. Finitary Winning in ω -regular Games. *ACM Transactions on Computational Logic*, 11(1), 2009.
- Cho74. Y. Choueka. Theories of automata on ω -tapes: A simplified approach. *Journal of Computer and System Sciences*, 8:117–141, 1974.
- EJ88. E. Allen Emerson and Charanjit S. Jutla. The Complexity of Tree Automata and Logics of Programs. In *Proceedings of the 29th Annual Symposium on Foundations of Computer Science, FoCS'88*, pages 328–337. IEEE Computer Society, 1988.
- GH82. Yuri Gurevich and Leo Harrington. Trees, Automata, and Games. In *Proceedings of the 14th Annual ACM Symposium on Theory of Computing, STOC'82*, pages 60–65. ACM Press, 1982.
- HST10. Szczepan Hummel, Michal Skrzypczak, and Szymon Torunczyk. On the topological complexity of $\text{mso}+\text{u}$ and related automata models. In *International Symposium on Mathematical Foundations of Computer Science, MFCS'*, pages 429–440, 2010.
- MP92. Zohar Manna and Amir Pnueli. *The Temporal Logic of Reactive and Concurrent Systems: Specification*. Springer-Verlag, 1992.
- MS72. A. R. Meyer and L. J. Stockmeyer. The equivalence problem for regular expressions with squaring requires exponential space. In *Annual Symposium on Foundations of Computer Science, FoCS'*, pages 125–129. IEEE, 1972.

- PR89. Amir Pnueli and Roni Rosner. On the Synthesis of a Reactive Module. In *Proceedings of the 16th Annual ACM Symposium on Principles of Programming Languages, POPL'89*, pages 179–190, 1989.
- RW87. Peter J. Ramadge and W. M. Wonham. Supervisory control of a class of discrete-event processes. *SIAM Journal on Control and Optimization*, 25(1):206–230, 1987.
- Saf92. S. Safra. Exponential Determinization for ω -Automata with Strong-Fairness Acceptance Condition. In *Annual ACM Symposium on Theory of Computing, STOC'*. ACM Press, 1992.
- Tho97. Wolfgang Thomas. Languages, Automata, and Logic. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages*, volume 3, Beyond Words, chapter 7, pages 389–455. Springer, 1997.
- Wad84. W. W. Wadge. *Reducibility and Determinateness of Baire Spaces*. PhD thesis, UC Berkeley, 1984.